



'Turbo' coding for deep space applications

Andersen, Jakob Dahl

Published in:
Proceedings of the IEEE International Symposium on Information Theory

Link to article, DOI:
[10.1109/ISIT.1995.531138](https://doi.org/10.1109/ISIT.1995.531138)

Publication date:
1995

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Andersen, J. D. (1995). 'Turbo' coding for deep space applications. In *Proceedings of the IEEE International Symposium on Information Theory* (pp. 36-36). IEEE. <https://doi.org/10.1109/ISIT.1995.531138>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

'Turbo' Coding for Deep Space Applications

Jakob Dahl Andersen (jda@it.dtu.dk)

Institute of Telecommunication, Technical University of Denmark, DK-2800 Lyngby, Denmark

Abstract - The performance of the 'turbo' coding scheme is measured and an error floor is discovered. These residual errors are corrected with an outer BCH code. The complexity of the system is discussed, and for low data rates a realizable system operating at E_b/N_0 below 0.2 dB is presented.

I. INTRODUCTION

Recently it has been discovered that a very good performance can be achieved with iterative decoding of a parallel concatenation of small convolutional codes [1]. This coding scheme is named 'turbo' coding. The basic idea is to encode the information sequence twice, the second time after a pseudo-random interleaver, and to do iterative decoding on the two encoded sequences in two decoders. The system can be regarded as a kind of product code. Due to the information exchange among the two decoders the decoding algorithm must provide soft output. We use the MAP algorithm [2] which actually calculates the a posteriori probability of each information bit. The convolutional codes are used in a recursive systematic form since it gives an improved performance with this system.

II. THE ERROR FLOOR

The first simulations were based on the recursive systematic code $(1, 1+D^4/1+D+D^2+D^3+D^4)$. We use the same code for both encoders but for the second one the information sequence is not transmitted. This gives an overall rate of 1/3. We use a block length of 10384 information bits. For all simulations presented in this paper all numbers including the channel input are represented as floating point values.

As seen from Figure 1, the results achieved with this system are very promising since the Bit Error Rate (BER) after 18 iterations is close to 10^{-5} already at 0.2 dB. Unfortunately the BER decreases very slowly for improved SNR. What we see are many frames with only a few bit errors. This is due to the low free distance of this coding scheme. The free distance of this system might be as low as 10. The actual profile depends on the specific interleaver.

A search for better interleavers might give improved performance. However, the main problem is combinations of two low weight words for the basic code. Consequently the performance with interleaver structures like block interleavers is quite poor, and a search among the random interleavers can only remove a couple of the worst low weight patterns.

III. THE EXTENDED 'TURBO' CODING SCHEME

An obvious way to remove the error floor (or saddle) is to use an outer code. Since the bursts consist of very few bit errors, we will use a (10384,10000) BCH code capable of correcting 24 errors. This outer code corrects all the residual errors, but we loose 0.16 dB due to the decreased rate. With this system the Probability of Frame Loss (PFL) is below 10^{-4} at 0.4 dB.

Improved performance can be achieved with a system based on rate 1/3 codes with only 8 states. This gives rate 1/5 for the 'turbo' coding scheme. In this case we have also used the outer BCH code.

With this system we have simulated 25,000 frames without frame loss at 0.1 dB. This means that the 90% confidence level for the PFL is below 10^{-4} . The BER is shown in Figure 1.

IV. COMPLEXITY

The performance must of course be compared to the complexity. We have estimated the number of operations needed in the MAP algorithm for recursive systematic codes and conclude that this is about 4 times the number of operations in a Viterbi decoder. This means that the number of operations for 18 iterations with $M=3$ codes is in the order of 2^{12} . We believe that with a logarithm quantization an 8 bit representation is sufficient for the internal representation in the MAP decoder. With this quantization and channel input quantized in 16 levels we expect a performance degradation about 0.1 dB.

For low data rates the 'turbo' coding scheme can be implemented with only one MAP decoder (used 2×18 times), and the decoder for the BCH code can be implemented on a DSP. Further the calculations inside the MAP decoder can be serialized, using the same hardware for each state.

This means that for data rates below 100 kbit/s the complexity of this system is moderate, and the extended 'turbo' coding scheme might be an alternative to ordinary concatenated systems.

REFERENCES.

- [1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error-correcting Coding and Decoding: Turbo-codes(1)", *Proc. ICC '93*, May 1993, pp. 1064-1070.
- [2] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", *IEEE Transactions on Information Theory*, Vol. IT-20, March 1974, pp. 284-287.

